# Max-Plus Algebraの数式処理

AUTHOR(S):

金, 龍; 金, 園益; 伊藤, 雅明

CITATION:

金, 龍 ...[et al]. Max-Plus Algebraの数式処理. 数理解析研究所講究録 2003, 1335: 6-12

ISSUE DATE:

2003-07

URL:

http://hdl.handle.net/2433/43332

数理解析研究所講究録 1335 巻 2003 年 6-12



# Max-Plus Algebraの数式処理

金　龍
LONG JIN
広島大学大学院　工学研究科
GRADUATE SCHOLL OF ENGINEERING, HIROSHIMA UNIVERSITY*

金　園益
YUANYI JIN
広島大学大学院　工学研究科
GRADUATE SCHOLL OF ENGINEERING, HIROSHIMA UNIVERSITY†

伊藤　雅明
MASAAKI ITO
広島大学大学院　工学研究科
GRADUATE SCHOLL OF ENGINEERING, HIROSHIMA UNIVERSITY‡

## 1　はじめに

ソリトン方程式と呼ばれるあるクラスの非線形偏微分方程式には無限個の保存量や、$N$ 個の孤立波の相互作用を表す $N-$ ソリトン解が存在することが知られている [1][2]。このような性質は微分方程式だけに限らず、独立変数を離散化した差分方程式や従属変数をも離散化した超離散方程式にも存在する。

しかし、max-plus algebra 演算を含む超離散系においては、その計算の複雑性が研究の発展を阻害しており、これを取り除くことが重要な課題の一つである。ソリトン研究の進歩に数式処理の利用が大きく貢献したことを考えると、超離散方程式に現れる max-plus algebra を数式処理化することは超離散方程式の研究に貢献できるものと考えられる。

そこで、本稿では超離散方程式のソリトン解を求めるために必要な max-plus algebra [3] の基本演算、超離散方程式の標準形への書き換え、パラメータに対する条件下での簡約化等を行うためのアルゴリズムを示す。また、その一部を数式処理システム REDUCE 上に実装し、その有効性を示す。

*kinryu@amath.hiroshima-u.ac.jp
†yyjin@amath.hiroshima-u.ac.jp
‡ito@amath.hiroshima-u.ac.jp

## 2 ソリトン方程式の離散化

生存競争のモデルの一つであるロトカーボルテラ（Lotka-Volterra）方程式

$$\frac{dv_n}{dt} = v_n(v_{n-1} - v_{n+1}) \tag{1}$$

を例にとり、ソリトン方程式の離散化及びその解の構造を見ておく。ここで、$v_n(t)$ は第 $n$ 種の生物の個体数を表している。ロトカーボルテラ方程式（1）は従属変数変換

$$v_n = \frac{f_{n-1}f_{n+2}}{f_n f_{n+1}}$$

によって $v_n$ から $f_n$ に移ると、その二つの孤立波の相互作用を表す 2− ソリトン解は

$$\begin{cases} f_n = 1 + \exp\eta_1 + \exp\eta_2 + a_{12}\exp(\eta_1+\eta_2)\,, \\ \eta_i = k_i n - \omega_i t + \eta_i^0 \quad (i=1,2)\,, \\ \omega_i = \sinh k_i \quad (i=1,2)\,, \\ a_{12} = \sinh^2\frac{k_1-k_2}{2} / \sinh^2\frac{k_1+k_2}{2} \end{cases} \tag{2}$$

で与えられる。

さらに、次のように時間を離散化した差分方程式を考える。

$$\frac{u_n^{t+1} - u_n^t}{\delta} = u_n^t u_{n-1}^t - u_n^{t+1}u_{n+1}^{t+1} \tag{3}$$

この方程式は差分ロトカーボルテラ方程式と呼ばれている。ここで、$\delta$ は時間の差分間隔であり、$u_n^{t+1}$ は $u_n(t+\delta)$ を表している。この差分ロトカーボルテラ方程式は、もちろん $\delta \to 0$ の極限では元のロトカーボルテラ方程式と一致する。差分ロトカーボルテラ方程式 (3) を同様の従属変数変換

$$u_n^t = (f_{n-1}^t f_{n+2}^{t+1})/(f_n^t f_{n+1}^{t+1})$$

によって $u_n^t$ から $f_n^t$ へ移ると、その 2 −ソリトン解は

$$\begin{cases} f_n = 1 + \exp\eta_1 + \exp\eta_2 + a_{12}\exp(\eta_1+\eta_2)\,, \\ \eta_i = k_i n - \omega_i t + \eta_i^0 \quad (i=1,2)\,, \\ \omega_i = \log\frac{1+\delta(1+e^{k_i})}{1+\delta(1+e^{-k_i})} \quad (i=1,2)\,, \\ a_{12} = \sinh^2\frac{k_1-k_2}{2} / \sinh^2\frac{k_1+k_2}{2} \end{cases} \tag{4}$$

で与えられ。これらの解は分散関係を除いて元のロトカーボルテラ方程式と同じ構造をしている。見方を変えれば、上記の離散化は解の構造を保存する離散化とも言える。差分ロトカーボルテラ方程式 (3) 式に対して、$u_n^t = \exp(U_n^t/\epsilon), \delta = \exp(-1/\epsilon)$ とし、さらに $\epsilon \to +0$ の極限をとると、次のような方程式

$$U_n^{t+1} - U_n^t = \max(0, U_{n-1}^t - 1) - \max(0, U_{n+1}^{t+1} - 1)$$

が得られる。この方程式は初期条件として $U_n^0$ ($n$ は整数 ) をすべて整数にと取れば、任意の時刻の $U_n^t$ もすべて整数で表されることを示しており、従属変数まで離散化される。このことより、この方程式は超離散ロトカーボルテラ方程式と呼ばれている。

方程式を超離散化した同じ手法 [4][5] を差分ロトカーボルテラ方程式の解 (4) に適用すれば、従属変数変換は

$$U_n^t = F_{n-1}^t + F_{n+2}^{t+1} - F_n^t - F_{n+1}^{t+1} \tag{5}$$

と表され、2－ソリトン解は

$$\begin{cases} F_n^t = \max(0, \Xi_1, \Xi_2, \Xi_1 + \Xi_2 + A)\ , \\ \Xi_i = K_i n - \Omega_i t + \Xi_i^0 \quad (i = 1, 2)\ , \\ \Omega_i = \max(0, K_i - 1) - \max(0, -K_i - 1) \quad (i = 1, 2)\ , \\ A = |K_1 - K_2| - |K_1 + K_2| \end{cases} \tag{6}$$

となる。

超離散方程式の解では、微分又は差分方程式の解に現れていている各項が max の要素として表現される。それ故、$N$-ソリトン解の max の要素の数は $2^N$ 個になり、その各要素が上記のような phase や分散関係を持っていると、その計算は非常に複雑なものとなる。広田の双線形形式の立場からすれば、2－ソリトン解を持つ方程式は容易に作ることができるが、それらがすべて可積分系であるとは限らない。ところが、今まで知られている多くの可積分系の例からすると、3－ソリトン解を持つことが可積分であるかを判断する一つの目安になっている。その意味でも、与えられた超離散方程式が 3－ソリトン解を持つことを検証することは重要である。しかしながら、これらの計算は非常に複雑で、もはや人間の手におえるものではなく、計算機による数式処理の利用が不可欠である。

## 3 超離散方程式の標準化

max 演算を含む超離散方程式を解くためには

1. 超離散方程式の標準形への書き換え
2. max の要素の簡約化

の二つの操作が必要である。

ここで、標準化とは、max-plus algebra を用いて超離散方程式の左辺と右辺を一つの max 演算で表現することであり、標準化された式を標準形と呼ぶことにする。標準化を行うためには max-pulas algebra の次の二つの基本演算を用いる。

基本演算 1 : $\max(a + b) + c = \max(a + b + c)$
基本演算 2 : $\max(\max(a, b), c) = \max(a, b, c)$

ただし、max-plus algebra では max のマイナス演算は定義されてないので、マイナスを含む方程式に対してはマイナスを含まないように方程式を書き直す必要がある。基本演算により、max の和で表される式及び max の入れ子になっている式を一つのの max 演算にまとめ、標準形にする。数式処理システム REDUCE 上でこれをを実現するプログラムの関数名を amp とする。これの実行により、

$$\mathrm{amp}(\max(\cdots\ \cdots)_1 + \max(\cdots\ \cdots)_2 + \max(\cdots\ \cdots)_3 + \cdots) = \max(\cdots\ \cdots)_f$$

となる。ただし、左辺の $(\cdots\ \cdots)_i$ $(i = 1, 2, 3, \cdots)$ には max 演算が含まれていても構わないが、右辺の $(\cdots\ \cdots)_f$ には含まれない。

次に、標準化を行うための超離散方程式の書き換えを行う。max-plus algebra では max のマイナス演算は定義されてないので、標準化を行うためには与えられた式にマイナス演算が含まれないように書き換え

る必要がある。例えば、超離散ロトカーボルテラ方程式

$$U_n^{t+1} - U_n^t = \max(0, U_{n-1}^t - 1) - \max(0, U_{n+1}^{t+1} - 1)$$

に変換式 $U_n^t = F_{n-1}^t + F_{n+2}^{t+1} - F_n^t - F_{n+1}^{t+1}$ を代入すると、

$$\begin{aligned} & F_{n-1}^{t+1} + F_{n+2}^{t+2} - F_n^{t+1} - F_{n+1}^{t+2} \\ - \quad & (F_{n-1}^t + F_{n+2}^{t+1} - F_n^t - F_{n+1}^{t+1}) \\ = \quad & \max(0, F_{n-2}^t + F_{n+1}^{t+1} - F_{n-1}^t - F_n^{t+1} - 1) \\ - \quad & \max(0, F_n^{t+1} + F_{n+3}^{t+2} - F_{n+1}^{t+1} - F_{n+2}^{t+2} - 1) \end{aligned}$$

となり、マイナス演算が含まれる。そこで、マイナス演算の項を移項し、基本演算を用いて両辺を一つの max 演算にまとめると、

$$\begin{aligned} & \max(F_{n-1}^{t+1} + F_{n+2}^{t+2} + F_n^t + F_{n+1}^{t+1} + 1, \\ & F_n^{t+1} + F_{n+3}^{t+2} + F_{n-1}^{t+1} + F_n^t) \\ = \quad & \max(F_{n-1}^t + F_{n+2}^{t+2} + F_n^{t+1} + F_{n+1}^{t+2} + 1, \\ & F_{n-2}^t + F_{n+1}^{t+1} + F_{n+1}^{t+2} + F_{n+2}^{t+1}) \end{aligned} \tag{7}$$

が得られる。(7) 式の $F_{n+a}^{t+b}(a = 0, \pm 1, \pm 2, b = 0, 1, 2)$ には、(6) 式で与えられるように max の演算が入れ子の形で含まれているので、更なる標準化の操作が必要である。max のマイナス演算を含まない (7) 式に対して、次のように標準化を行う。

まず、超離散ロトカーボルテラ方程式の解 $F_n^t = \max(0, \Xi_1, \Xi_2, \Xi_1 + \Xi_2 + A)$ に対して、phase

$$\Xi_j = K_j n - \Omega_j t + \Xi_j^0 \quad (i = 1, 2)$$

を代入する。ここで

$$\begin{cases} x_1 = K_1 n - \Omega_1 t + \Xi_1^0 \\ x_2 = K_2 n - \Omega_2 t + \Xi_2^0 \end{cases},$$

とおくと、(7) 式に現れている $F_{n+a}^{t+b}$ は

$$F_{n+a}^{t+b} = \max(0, x_1 + K_1 a - \Omega_1 b, x_2 + K_2 a - \Omega_2 b, (x_1 + x_2) + (K_1 + K_2)a - (\Omega_1 + \Omega_2)b + A)$$

となる。REDUCE 上でこの $F_{n+a}^{t+b}$ を関数 $G(a, b)$ と定義すると、(3.1) 式の左辺（L）及び右辺（R）は次のようになる。

$$L = \max(G(-1,1) + G(2,2) + G(0,0) + G(1,1) + 1, G(0,1) + G(3,2) + G(-1,1) + G(0,0))$$

$$R = \max(G(-1,0) + G(2,2) + G(0,1) + G(1,2) + 1, G(-2,0) + G(1,1) + G(1,2) + G(2,1))$$

ここで REDUCE 上で定義した関数 amp を用いて $L$ と $R$ を標準化すると

$$L = \max(l_1, l_2, l_3, \cdots)$$

$$R = \max(l_1', l_2', l_3', \cdots)$$

となる。ここで $l_i$ $(l_i')$ $(i = 1, 2, \cdots)$ は

$m_1 x_1 + m_2 x_2 + m_3 K_1 + m_4 K_2 + m_5 \Omega_1 + m_6 \Omega_2 + m_7 A$　　（$m_i$ は整数。）

の形をしている。$\Omega_1$, $\Omega_2$ の中には max 演算が含まれているが、この段階ではこれ以上の標準化は行わない。
$F_n^t$ が超離散方程式 (7) の解であることを示すには、任意の $x_1, x_2$ に対して $L = R$ を示さなくてはならない。即ち、$x_1, x_2$ の同じ係数を持つ項のリストの max が等しくなければならない。そこで $l_i$ ($l_i^{'}$) を次のように書き表すことにする。

$$m_1x_1 + m_2x_2 + m_3K_1 + m_4K_2 + m_5\Omega_1 + m_6\Omega_2 + m_7A$$

$$\rightarrow ((m_1, m_2), (m_3K_1 + m_4K_2 + m_5\Omega_1 + m_6\Omega_2 + m_7A))$$

この変換を REDUCE 上で行う関数を getm1m2list とする。getm1m2list の実行により、L と R は

$$L = \max(\cdots, \cdots, ((m_1, m_2), (h_1, h_2, \cdots)), \cdots)$$

$$R = \max(\cdots, \cdots, ((m_1, m_2), (h_1^{'}, h_2^{'}, \cdots)), \cdots)$$

となる。ここで $h_i$ ($h_i^{'}$) は $m_3K_1 + m_4K_2 + m_5\Omega_1 + m_6\Omega_2 + m_7A$ の形を持つ。両辺の同じ $(m_1, m_2)$ のリストに対して

$$\max(h_1, h_2, h_3, \cdots) = \max(h_1^{'}, h_2^{'}, h_3^{'}, \cdots)$$

が示されれば $F_n^t$ が解であることが分かる。これを示すためには、max の中の要素の大小関係を用いて不要な項を取り除くための簡約化の操作が必要になる。

## 4 簡約化

$\max(h_1, h_2, h_3, \cdots)$ の中の $h_i (i = 1, 2, \cdots)$ には $\Omega_1$, $\Omega_2$, $A$ が含まれており、これらの条件を代入しないと完全な簡約化は行えない。しかし、最初からこれらの条件を付加すると計算量が増加するため、まずは、これらの条件付けない簡約化（無条件簡約化）を行う。$(h_1, h_2, \cdots)$ のリストからすべての $h_i, h_j$ $(i \leq j)$ の組を取り出し、それらに対して $h_i - h_j = C$ ($C$ は定数) となる組があれば、$h_i, h_j$ の大小関係が確定され、小さい方をリストから除くことができる。この簡約アルゴリズムを amp 関数に付け加える。

次に $\Omega_1, \Omega_2$ 及び $A$ に対する条件を入れた時の簡約化を考える。超離散ロトカーボルテラ方程式の分散関係式は

$$\Omega_j = \max(0, K_j - 1) - \max(0, -K_j - 1) \quad (j = 1, 2)$$

で与えられるので、$K_j$ を次のような条件

$$K_j \leq -1 \; ; -1 < K_j < 1 \; ; 1 \leq K_j \quad (j = 1, 2)$$

に分けて考えると、$\Omega_1, \Omega_2$ の値が $K_1, K_2$ で評価できる。
次に、

$$A = |K_1 - K_2| - |K_1 + K_2|$$

を決めるために、次のような条件

$$K_1 \geq K_2 (\text{または} K_1 \leq K_2)$$

に分けて考えると、$A$ の値が $K_1, K_2$ で評価できる。

これらの各々の条件の下で評価された $\Omega_1, \Omega_2$ 及び $A$ を代入すると、各リストの項が $K_1, K_2$ のみを含む

多項式になる。REDUCE 上でこれを実現する関数を getk1k2list とする。関数 getk1k2list の実行により、リストの各項は

$$m_3K_1 + m_4K_2 + m_5\Omega_1 + m_6\Omega_2 + m_7A + C$$

の形から

$$m_3^{'}K_1 + m_4^{'}K_2 + C$$

の形になる。次に、評価された $(h_1, h_2, \cdots)$ のリストからすべての $h_i, h_j$ $(i \leq j)$ の組を取り出し、次の演算行う。

$$h_i - h_j = n_1(K_1 - K_2) + n_2K_2 \qquad (n_1, n_2\text{は整数})$$

この時、$K_1$ と $K_2$ の仮定条件及び $n_1$ と $n_2$ 正負関係より $h_i, h_j$ の大小関係が判断できれば、小さい方をリストから除くことができる。REDUCE 上でこれを実現する関数を getlist とする

下記の仮定条件

$$\begin{cases} K_1 > 1, \quad K_2 > 1\ , \\ K_1 > K_2 \end{cases}$$

の下で、getlist の実行により L と R は共に次のように簡約化できる。

```
(
(4,4),
    (-2*K1 - 10*K2 + 1),
(3,4),
    (-8*K2 + 1),
(3,3),
    (- K1 - 5*K2),
(2,4),
    (-6*K2 + 1),
(2,3),
    (K1 - 4*K2),
(2,2),
    (- K1 - 2*K2),
(1,4),
    (K1 - 4*K2),
(1,3),
    (- K1 - 2*K2 + 1),
(1,2),
    (1,
    K1 - K2),
(1,1),
    (K1),
)
```

上記の結果から (1,2) のリスト以外は完全簡約化されたことがわかる。(1,2) のリストは完全簡約化はされてはいないが、項の数は 2 個しか残らず、人間の能力ですぐ判断できる。このように簡約化プログラムは解の検証に役立つことが示された。

今後の課題としては、与えられた条件式（分散関係等）からパラメータに対する場合分けを自動化することが残されている。

## 参 考 文 献

[1] R. K. Bullough and P. J. Caudray eds. ,*"Solitons"*, Springer Topics in Current Physics ,**17** (1980), New York.

[2] 広田　良吾,「直接法によるソリトンの数理」, 岩波書店 (1992).

[3] F.L. Baccelli, G. Cohen, G.J. Olsder and J.P. Quadrat, Synchronization and Linearity, (JOHN WILEY & SONS, 1992, New York).

[4] R. Hirota and S. Tsujimoto, Conserved Quantities of a Class of Nonlinear Difference-Difference Equations, J. Phys. Soc. Jpn. **64**(1995) 3125.

[5] T. Tokihiro, D. Takahashi and J. Satsuma, From Soliton Equations to Integrable Cellular Automata through a Limiting Procedure, Phys. Rev. Lett. **76**(1996) 3247.